

## Does CASE failure mean Entity Model failure?

---

There have been a number of references to the failure of CASE tools to measure up to their promise. A recent article in the Bill Pike column co-authored by Nicholas Wybolt (Enterprise Architect for McDonald's Corporate Enterprise Computing Architecture and head of McDonald's Advanced Technology Laboratory.) states "The shortfall and abject failure of CASE to deliver on its promise to help companies greatly improve on quality and productivity left a significant void". This remark is made in the context of Rapid Application Development trends, in which the effectiveness of integrated development environments (IDEs) like Delphi and Visual Basic is emphasised. These IDEs have not been integrated with CASE tools, and so there was no productive bridge between them. Those using the IDEs tend to use only the IDE, and no CASE tool, and because of the power of the IDEs they are very effective.

My impression is that in South Africa there are a number of organizations who HAVE achieved significant productivity gains from the use of CASE tools (in those cases where the CASE tool generates the code of the system).

This article focuses rather on those organizations that have decided CASE tools are not worth the effort. Have they also discontinued the production of entity models? This would be equivalent to throwing the baby out with the bath water.

I build systems using Delphi and JBuilder (two of the leading IDEs) and for every system I produce, the first action I take (after scoping the system) is to build an Entity Model. I have spoken to many others who are in agreement that the Entity Model forms the basis for rapid and correct development of the system functionality. In fact it is the entity model that makes it possible to launch directly into prototyping the GUI interfaces. The entity model is the infrastructure, which binds the prototype components sensibly together.

The entity model is the right vehicle for describing the semantics of the business. To attempt this at the level of the relational tables is not a sensible option since the relational theory was never effective for this purpose. To leave out the semantics altogether is to court disaster, since this is where the business understanding and the business rules are easily identified.

The conclusion is that even if you are one of those who believe that CASE has failed, **Entity Models remain an essential deliverable for the production and maintenance of information systems**. This means that you might choose to continue using your CASE tool just for this purpose.

*Any comments would be welcome.*  
Email [mikeking@MikeKingCourses.com](mailto:mikeking@MikeKingCourses.com)